Lecture 23

Probabilistically Checkable

Proof Systems (cont.)
linear function: \( \forall x, y \ f(x) + f(y) = f(x+y) \)

self-correcting:

\[
\text{if } f \text{ is } \frac{1}{8} \text{-close to linear } g
\]

- Do \( O(\log \frac{1}{\beta}) \) times
  - Pick \( y \) randomly
  - \( \text{answer}_i \leftarrow f(y) + f(x-y) \)
  - Output most common answer

self-testing: Given \( f \)

- Do \( O(\frac{\varepsilon}{\beta}) \) times
  - Pick \( x, y \) randomly
  - If \( f(x) + f(y) \neq f(x+y) \) Fail
  - Pass

- then \( \forall x, \ Pr[\text{output} = g(x)] \geq 1 - \beta \)

- if \( f \) linear, passes
  - if \( f \) \( \varepsilon \)-far from linear, fails
Probabilistically Checkable Proofs

\[ \text{Input } x \]

\[ \text{random string} \]

\[ \text{Proof } \Pi \]

---

Theorem you want to prove for today: \( \mathbf{X} \) is 3CNF

Thm \( \mathbf{X} \) is satisfiable

---

1) \( \forall x \in L \exists \Pi \text{ s.t. } \Pr_{r \in \{0,1\}^n} \left[ V, \Pi \text{ accepts} \right] = 1 \)

2) \( \forall x \in L \forall \Pi' \Pr_{r \in \{0,1\}^n} \left[ \forall V, \Pi' \text{ accepts} \right] \leq \frac{1}{4} \)
e.g. \( \text{SAT} \in \text{PCP}(0, n) \) ← proof settings of all \( n \) vars \( V \) doesn't need any randomness

\[ \text{Today:} \quad \text{NP} \leq \text{PCP} \left( O(n^3), O(1) \right) \]

\[ \text{Actually:} \quad \text{NP} \leq \text{PCP} \left( O(\log n), O(1) \right) \]

Let's start with a "warm-up".
\[ X \cdot y = \sum x_i y_i \quad \text{"inner product"} \]

\[ X \times y = (x_1 y_1, x_1 y_2, \ldots, x_n y_n) \quad \text{"outer product"} \]

**Fact.** if \( \overline{a} + \overline{b} \) then \( \Pr \left[ \overline{a} \cdot \overline{r} \neq \overline{b} \cdot \overline{r} \right] \geq \frac{1}{2} \)

if \( A \cdot B \neq C \) then \( \Pr \left[ A \cdot B \cdot \overline{r} \neq C \cdot \overline{r} \right] = \frac{1}{2} \)

A \cdot (B \cdot \overline{r}) \text{ take } O(n^2) \text{ to compute}
Fact: if \( \bar{a} + \bar{b} \) then \( \Pr_{\bar{r} \in \mathbb{R}^n} \left( \bar{a} \cdot \bar{r} \neq \bar{b} \cdot \bar{r} \right) \geq \frac{1}{2} \)

if \( A : B = C \) then \( \Pr_{\bar{r}} \left[ A : B \cdot \bar{r} \neq C \cdot \bar{r} \right] \geq \frac{1}{2} \)

Example "application": setting: given vector \( \bar{a} = (a_1, a_2, \ldots, a_n) \)

in one step: • can query \( a_i \)

• can specify \( \bar{y} \) or query \( \bar{a} \cdot \bar{y} \)

to test if \( \bar{a} = (0, 0, \ldots, 0) \):

Do several times:

pick \( \bar{r} \in \mathbb{R}^n \)

if \( \bar{a} \cdot \bar{r} \neq 0 \) output "Fail"

Output PASS

behavior: if \( \bar{a} = (0, \ldots, 0) \) will always PASS

if \( \bar{a} \neq (0, \ldots, 0) \) then FACT \( \Rightarrow \Pr_{\bar{r}} \left[ \bar{a} \cdot \bar{r} \neq 0 \right] \geq \frac{1}{2} \)

\( \Rightarrow \mathcal{O}(1) \) query \( 0 \)-testing algorithm for \( n \)-bit vector

in strange model
Arithmetization of 3SAT:

Boolean formula $F$ \iff arithmetic formula $A(F)$ over $\mathbb{Z}_2$

$T \iff 1$

$F \iff 0$

$x_i \iff x_i$

$\overline{x}_i \iff 1 - x_i$

$\lambda \land \beta \iff \lambda \cdot \beta$

$\lambda \lor \beta \iff 1 - (1 - \lambda) (1 - \beta)$

$\alpha \lor \beta \lor \gamma \iff 1 - (1 - \alpha) (1 - \beta) (1 - \gamma)$

Example: $x_1 \lor \overline{x}_2 \lor x_3 \iff 1 - (1 - x_1) (1 - x_2) (1 - x_3)$

Key point: $F$ satisfied by assignment $\alpha$ \iff $[A(F)](\alpha) = 1$
\[ F = \bigwedge C_x \quad \text{s.t.} \quad C_x = (y_{i_1} \lor y_{i_2} \lor y_{i_3}) \]

where \( y_{i_j} \in \{ x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n \} \)

Consider \( C^0(x) = (\hat{C}_1(x), \hat{C}_2(x), \ldots) \)

\[ \hat{C}_x(x) = \text{complement of arithmeticization of clause } C_x \]

\( \Rightarrow \) evaluates to 0 if \( x \) satisfies \( C_x \)

\( \Rightarrow C^0(x) = (0, \ldots, 0) \) if \( x \) satisfies \( F \)

Observe:
(1) each \( \hat{C}_x \) is \( \deg \leq 3 \) poly in \( x \)

(2) \( V \) knows coeffs of each \( \hat{C}_x \)

Need to convince \( V \) that \( C^0(a) = (\hat{C}_1(a), \hat{C}_2(a), \ldots) = (0, \ldots, 0) \) without sending assignment a
High level idea: special encoding of assignment

Encode satisfiability of \( F \) as a collection of polys in vars of assignment
- one for each clause
- eval to 0 if assignment satisfies clause
- low degree
- \( V \) knows coeffs - depend on structure of clause \& vars of clause.

Note: We are only concerned that \( V \) is poly time, note that solving SAT in poly time would be impressive.

However, want \( \pm 1 \) queries to proof to be constant
Idea for proof:

- Proof contains $C(a) \cdot r$ for $r \in \{0, 1\}$
- If $\forall i$, $\hat{C}_i(a) = 0$, $\Pr[C(a) = 0] = 1$
- If $\exists i$ s.t. $\hat{C}_i(a) \neq 0$, $\Pr[C(a) = 0] = \frac{1}{2}$
  \[ \Pr[C(a) = 1] \]

\[ F = \left\{ \text{all } C_i \text{ s.t. } C_i = (y_{i_1}, y_{i_2}, \ldots, y_{i_n}) \right\} \]
where $y_{i_j} \in \{0, 1\}$, $x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n$

$C(a) = (\hat{C}_1(a), \hat{C}_2(a), \ldots) \overset{?}{=} (0, 0, 1, \ldots)$

$\Rightarrow \mod 2$ arithmetic

$\Rightarrow \text{mod } 2$ arithmetic

What does $C(a) \cdot r$ look like?

\[ \sum_{i} r_i C_0(a) = \Gamma + \sum_{i} a_i \hat{x}_i + \sum_{i,j} a_i a_j \frac{\beta_{ij}}{x_{ij}} + \sum_{i,j,k} a_i a_j a_k \gamma_{ijk} \text{ (mod 2)} \]

From here on:
- $x_i \rightarrow x_i$
- $\beta_{ij} \rightarrow y_{ij}$
- $y_{ijk} \rightarrow z_{ijk}$

$\forall C_i$ depends on $\hat{C}_i$ and coefs in $\hat{C}_i$
High level idea: Special encoding of assignment

- Proof writes out all linear ftns of assignment
  - deg 2,
  - deg 3

- Possible "confusion": "symmetric" for linear case

\[ f_x(a) = x \cdot a = A_a(x) \]

\[ \text{inner product} \]

- for deg 2,3:
  - \( B_a(y) = (a \cdot a)^T \cdot y \)
  - \( C_a(y) = (a \cdot a \cdot a)^T \cdot z \)

\( A_a, B_a, C_a \) are all linear ftns \( \Rightarrow \) can test linearity & self-correct

Proof can cheat! - what if \( A_u, B_a, C_a \) come from different assignments

- is a satisfying?
Complete description of truth tables of $A, B, C$ for all inputs $x, y, z$.

A = all linear fits

B = all deg 2 fits

$C = \text{deg } 3$ fits

We only care about their values at one input.

W.G.

These are the only care about their values at one input.

A

$A_n \rightarrow F_2$
\[ x \odot y = (x_1 y_1, x_1 y_2, x_2 y_3, \ldots, x_n y_n) \]

**Proof contains:**

- Complete description of truth tables of \( \tilde{A}, \tilde{B}, \tilde{C} \) for all inputs \( x, y, z \)
- Only need value at \( x = x, y = y, z = z \)
- But extra info helps us check consistency

---

### What does verifier need to check in proof? 

1. \( \tilde{A}, \tilde{B}, \tilde{C} \) in right form
   - All are linear fets
   - Correspond to same assignment \( a \)
   - i.e., \( \tilde{A}(x) = a^T x \implies \tilde{B}(y) = (a a)^T y \implies \tilde{C}(z) = (a o a o a)^T z \)
   - Test consistency of self-corrections

2. \( a \) is satisfying assignment
   - All \( \tilde{C}_x \)'s evaluate to 0 on \( a \)

\( \text{(recall} \quad C^o(a) = (\tilde{C}_1(a), \tilde{C}_2(a), \ldots)^T = (0, 0, \ldots) \text{)} \)
\[ X \cdot y = (x_1 y_1, x_2 y_2, \ldots, x_n y_n) \]

**Proof contains:**

Complete description of truth tables of \( A, B, C \) for all inputs \( x, y, z \)

- Only need value at \( x = x', y = y', z = z' \)
- But extra info helps us check consistency

---

**Test 1:** \( \hat{A}, \hat{B}, \hat{C} \) in right form \( \Rightarrow \) all are linear fans

- **Test:** \( \hat{A}, \hat{B}, \hat{C} \) are all \( \epsilon \)-close to linear (i.e., if all linear, PASS, if any one is \( \epsilon \)-far FAIL) in \( O(1) \) queries

- From now on, use self corrector to get

  \[ SC - \hat{A}, SC - \hat{B}, SC - \hat{C} \] for all inputs

\[ \Rightarrow \] Can only test \( \epsilon \)-close to linear but can self-correct to access the linear fans.

\[ \uparrow \quad \uparrow \quad \uparrow \]

- a
  - b
  - c

\[ a o a^2, a a a o a^2, a o a o a \]

\[ \Rightarrow \] unlikely to see error
\[ X^oY = (X_1Y_1, X_1Y_2, \ldots, X_nY_n) \]

**Proof contains:**

Complete description of truth tables of \( A, B, C \) for all inputs \( x, y, z \)

Only need value at \( x = x_1, y = y_1, z = z_1 \)

But extra info helps us check consistency

---

**Test (1)** \( A, B, C \) in right form: all are linear faiths

\[ \text{correspond to same assignment } a \]

\[ \text{i.e. } \tilde{A}(x) = a^T x \Rightarrow \tilde{B}(y) = (a\circ a)^T y \Rightarrow \tilde{C}(z) = (a\circ a\circ a)^T z \]

**Test consistency of self-corrections**

**Goal:** Pass if \( sc-\tilde{B} = sc-\tilde{A} \circ sc-\tilde{A} \)

\[ sc-\tilde{C} = sc-\tilde{A} \circ sc-\tilde{B} \]

**Outer Product Tester:** Pick random \( x_1, x_2, y \)

Test \( sc-\tilde{A}(x_1) \cdot sc-\tilde{A}(x_2) = \left[ \sum a_i x_{ik} \circ \sum a_i x_{jk} \right] = \sum a_i a_j x_{ik} x_{jk} \)

\[ = sc-\tilde{B}(x_1 o x_2) \]

\[ sc-\tilde{A}(x) \cdot sc-\tilde{B}(y) = \left[ \sum a_i x_{ik} \circ \sum b_j y_{jk} \right] = \sum a_i b_j x_{ik} y_{jk} \]

\[ = sc-\tilde{C}(x o y) \]

\( \Theta \) not uniformly distributed
**Definition:**

- $A : \mathbb{F}_2^n \to \mathbb{F}_2$, $A(x) = \sum a_i x_i = a^T x$

- $B : \mathbb{F}_2^n \to \mathbb{F}_2$, $B(y) = \sum a_i a_j y_{ij} = (a^o a)^T y$

- $C : \mathbb{F}_2^n \to \mathbb{F}_2$, $C(z) = \sum a_i a_j a_k z_{ijk} = (a^o a^o a)^T z$

**Proof contains:**

Complete description of truth tables of $A, B, C$ for all inputs $x, y, z$

Only need value at $x = x_1, y = y_1, z = z_1$

But extra info helps us check consistency.

**Test:**

$sc^{-1} A(x_1) \cdot sc^{-1} A(x_2) = \left( \sum a_i x_i \cdot \sum a_j x_j \right) = \sum a_i a_j x_i x_j = \sum b k x_i x_j$

Test passes ← since "blue" equalities hold

If $b = a \circ a$ ∧ test passes ← since "blue" equalities hold

If $b \neq a \circ a$:

$A(x_1) \cdot A(x_2) = B(x_1 \circ x_2) = b$

Diagram:

- Input $x_1 \circ x_2$
- Intermediate output $a \circ a$
- Final output $b$
if \( b = a \cdot a \cdot a \):

What is \( \text{prob} \)?

\[
\begin{align*}
\text{Fact} & \implies \Pr \left[ (a \cdot a) \cdot X_2 \neq b \cdot X_2 \right] = \frac{1}{2} \\
\text{if} \ (a \cdot a) \cdot X_2 \neq b \cdot X_2 & \implies \Pr \left[ X_1 : (a \cdot a) \cdot X_2 \neq X_1 \cdot b \cdot X_2 \right] = \frac{1}{2} \\
& \implies \Pr \left[ \text{fail test} \right] \geq \frac{1}{4}
\end{align*}
\]

So passing test

\( \implies \) safe to assume \( b = a \cdot a \cdot a \)!

Similarly passing after that

\( \implies \) safe to assume \( C = a \cdot a \cdot a \cdot a \)
\[ X \circ Y = (x_1 y_1, x_1 y_2, x_2 y_1, \ldots, x_n y_n) \]

**Proof contains:**

Complete description of truth tables of \(A, B, C\) for all inputs \(x, y, z\)

- Only need value at \(x = x', y = \beta, z = \gamma\)
- But extra info helps us check consistency

---

**Test (1)** \(A, B, C\) in right form:

- All are linear fets
- Correspond to same assignment \(a\)
  
  \(\hat{A}(x) = a^T x \Rightarrow \hat{B}(y) = (a \circ a)^T y \Rightarrow \hat{C}(z) = (a \circ a \circ a)^T z\)

**Goal:** Pass if \(sc^{-A} = sc^{-A} \circ sc^{-A}\)

**Outer Product Tester:**

Pick random \(x_1, x_2, x_y\)

Test \(sc^{-A}(x_1) \cdot sc^{-A}(x_2) = \left[ \sum a_i x_{ik} \circ \sum a_j x_{kj} \right] = \sum a_i a_j x_{ik} x_{kj} = \sum b_{ij} x_{ik} x_{kj}\)

\( = sc^{-B}(x_1 \circ x_2) \checkmark\)

Test \(sc^{-A}(x) \cdot sc^{-B}(y) = \left[ \sum a_i x_{ik} \circ \sum b_{ik} y_{jk} \right] = \sum a_i b_{ik} x_{ik} y_{jk} = sc^{-E}(x \circ y) \checkmark\)

\(\Theta = \text{not uniformly distributed}\)
\[ X^y = (x_1 y_1, x_1 y_2, \ldots, x_n y_1, \ldots, x_n y_m) \]

**Proof contains:**

- Complete description of truth tables of \( A, B, C \) for all inputs \( x, y, z \)
  
  - Only need value at \( x = \alpha, y = \beta, z = \gamma \)
  
  - But extra info helps us check consistency

\[ A : \mathbb{F}_2^n \rightarrow \mathbb{F}_2 \]
\[ A(x) = \sum a_i x_i = a^T x \]

\[ B : \mathbb{F}_2^n \rightarrow \mathbb{F}_2 \]
\[ B(y) = \sum_{i,j} a_{i,j} y_{i,j} = (a a^T) y \]

\[ C : \mathbb{F}_2^n \rightarrow \mathbb{F}_2 \]
\[ C(x) = \sum_{i,j,k} a_{i,j,k} x_{i,j,k} = (a a a^T) z \]

---

**Test 1:** \( A, B, C \) in right form: all are linear fets

- Test \( A, B, C \) are all \( \frac{1}{\gamma} \)-close to linear (i.e., if all linear, PASS, if any one is \( \frac{1}{\gamma} \)-far FAIL) in \( O(1) \) queries

- From now on, use self corrector to get
  
  \[ SC: A, SC: B, SC: C \] for all inputs

  - Use \( \beta \) = prob of getting wrong answer in SC that is so small (\( \leq \frac{1}{\text{polylogarithmic constant}} \)) that union bound over all queries to \( SC: A, SC: B, SC: C \)

  \[ \Rightarrow \text{unlikely to see error} \]
X \cdot y = (x_1 y_1, x_2 y_2, x_3 y_3, \ldots, x_n y_n)

\begin{align*}
A : \mathbb{F}_2^n &\to \mathbb{F}_2 \\
A(x) &= \sum a_i x_i = a^T x
\end{align*}

\begin{align*}
B : \mathbb{F}_2^n &\to \mathbb{F}_2 \\
B(y) &= \sum a_i y_i = (a \cdot a)^T y
\end{align*}

\begin{align*}
C : \mathbb{F}_2^n &\to \mathbb{F}_2 \\
C(z) &= \sum a_i z_i = (a \cdot a \cdot a)^T z
\end{align*}

Proof contains:

- Complete description of truth tables of \( A, B, C \) for all inputs \( x, y, z \)
- Only need value at \( x = 0, y = 0, z = 0 \)
- But extra info helps us check consistency

What does verifier need to check in proof?

1. \( A, B, C \) in right form
   - All are linear fets
   - Correspond to same assignment \( a \)
   - \( a \) is satisfying assignment
   - All \( \hat{a} \)'s evaluate to 0 on \( a \)

\( a \) is satisfying assignment

\( (\text{recall } C(a) = (\hat{C}_1(a), \hat{C}_2(a), \ldots) = (0, 0, \ldots) \)
How to do (2): 

1. Cell self-correctors ⇒ recover linear terms \( a_j, a_0a, a_0a_0a \)
2. \( \alpha \) represents assignment, but we don't know it
3. A satisfying \( \Rightarrow C^\alpha(a) = (\hat{\zeta}_1(a), \hat{\zeta}_2(a), \ldots) = (0, 0, \ldots, 0) \)

**Satisfiability Test:**

1. Pick \( r \in \mathbb{F}_2^n \)
2. Compute \( \alpha_i, \beta_{ij}, \gamma_{ijk} \)

3. Query proof to get

\[
\begin{align*}
\text{SC - } A (\alpha_1, \ldots, \alpha_n) &= w_0 \\
\text{SC - } B (\beta_1, \ldots, \beta_n) &= w_1 \\
\text{SC - } C (\gamma_1, \ldots, \gamma_{mn}) &= w_2
\end{align*}
\]

4. Verify \( 0 = r + w_0 + w_1 + w_2 \pmod{2} \)

5. Hopefully means \( \sum r_i \hat{\zeta}_i(a) = 0 \)

**Why do this?**

- If \( \forall i \hat{\zeta}_i(a) = 0 \)
  - \( \Pr[\text{pass}] = 1 \)
- If \( \exists i \) s.t. \( \hat{\zeta}_i(a) \neq 0 \)
  - Fact \( \Rightarrow \Pr[\sum r_i \hat{\zeta}_i(a) = 0] = \frac{1}{2^k} \)
  - So after \( k \) times
  - \( \Pr[\text{pass}] = \frac{1}{2^k} \)